

**AMENDMENT TO THE CLAIMS**

*The following claim listing replaces all prior listings and versions of the claims:*

**LISTING OF CLAIMS**

1. (Currently Amended) A program conversion device for a processor which has an instruction set including an instruction that waits for a predetermined response from an outside source when the instruction is executed, comprising:

a CPU; and

a compiler system including:

a loop structure transforming unit operable to perform double looping transformation so as to transform a structure of a loop, which is included in an input program and whose iteration count is x, into a nested structure where a loop whose iteration count is y is an inner loop and a loop whose iteration count is x/y is an outer loop; and

an instruction placing unit operable to convert the input program into an output program including the instruction by placing the instruction in a position outside the inner loop,

wherein the iteration count y of the input program in the inner loop is determined such that processing time of the input program in the inner loop constitutes all or part of latency time of the instruction placed outside the inner loop.

2. (Original) The program conversion device according to Claim 1,

wherein said loop structure transforming unit includes:

a loop detecting unit operable to detect a loop included in the input program;

an iteration count detecting unit operable to detect an iteration count of the detected loop;

a response wait cycle count detecting unit operable to detect the number of response wait cycles which is the number of cycles to wait for the predetermined response when the instruction is executed;

a cycles-per-sequence detecting unit operable to detect the number of cycles per sequence required for one set of iteration processing of the detected loop;

a loop splitting unit operable to split off, from the detected loop, a loop whose iteration count is derived from (the number of response wait cycles/the number of cycles per sequence);  
and

a double looping transforming unit operable to perform double looping transformation so as to build a nested structure where the loop whose iteration count is derived from (the number of response wait cycles/the number of cycles per sequence) is an inner loop and a loop whose iteration count is derived from (the iteration count of the detected loop/the iteration count of the inner loop) is an outer loop.

3. (Original) The program conversion device according to Claim 1, further comprising  
an optimization directive information receiving unit operable to receive optimization  
directive information which relates to optimization.

4. (Original) The program conversion device according to Claim 3,  
wherein said optimization directive information receiving unit is operable to receive a  
minimum iteration count of the loop included in the input program,  
said loop structure transforming unit is operable to, when an execution count of the loop  
is non-fixed, extract iteration processing having the minimum iteration count from the loop on

the basis of the minimum iteration count and to perform double looping transformation on the extracted iteration processing of the loop.

5. (Original) The program conversion device according to Claim 1,  
wherein the instruction is an instruction that has a possibility of causing an interlock.
6. (Original) The program conversion device according to Claim 5,  
wherein the instruction that has a possibility of causing an interlock is a prefetch instruction for prefetching data from main memory to a cache.
7. (Original) The program conversion device according to Claim 6, further comprising  
a scheduling unit operable to perform instruction scheduling,  
wherein said loop structure transforming unit is operable to split off, from the loop whose iteration count is  $x$ , a loop whose iteration count is  $y$  and which is executed corresponding to the number of cycles required to execute the prefetch instruction, based on a result obtained by said scheduling unit, and operable to perform double looping transformation so as to build a nested structure where the loop whose iteration count is  $y$  is an inner loop and a loop whose iteration count is  $x/y$  is an outer loop.

8. (Original) The program conversion device according to Claim 1,  
wherein after the instruction is executed, a plurality of cycles are required until a time comes when a predetermined resource will be referable.

9. (Original) The program conversion device according to Claim 8, wherein the instruction that requires the plurality is an instruction for accessing one of main memory and a cache.

10. (Original) The program conversion device according to Claim 1, wherein said loop structure transforming unit is operable to split off, from the loop whose iteration count is  $x$ , the loop whose iteration count is  $y$  and which is executed in accordance with an advance in a cache line size made by an address of an array referenced within the loop whose iteration count is  $x$ , and operable to perform double looping transformation so that the loop whose iteration count is  $y$  is an inner loop and the loop whose iteration count is  $x/y$  is an outer loop.

11. (Original) The program conversion device according to Claim 10, wherein when a plurality of arrays are present, said loop structure transforming unit is operable to further perform, in accordance with the number of the arrays, proportional dividing transformation to proportionally divide the loop whose iteration count is  $y$  and on which the double looping transformation has been performed.

12. (Original) The program conversion device according to Claim 11, wherein when sizes of array elements of the plurality of arrays are different, the loop whose iteration count is  $y$  is proportionally divided in the proportional dividing transformation in accordance with a ratio of the sizes.

13. (Original) The program conversion device according to Claim 11, wherein when each stride of the plurality of arrays is different, a stride referring to addresses advanced per set of the iteration processing of the loop, the loop whose iteration count is  $y$  is proportionally divided in the proportional dividing transformation in accordance with a ratio of the strides.

14. (Original) The program conversion device according to Claim 11, wherein when an inner loop is transformed, a conditional statement is generated for each divided loop and the proportional dividing transformation is performed so that each divided loop is executed within a same inner loop.

15. (Original) The program conversion device according to Claim 10, wherein when the loop whose iteration count is  $y$  is split off from the loop whose iteration count is  $x$  and a remainder  $z$  left over after a calculation of  $x/y$  is not zero, said loop structure transforming unit is operable to perform peeling processing and then double looping transformation on iteration processing that is to be executed  $z$  number of times.

16. (Original) The program conversion device according to Claim 15, wherein when the remainder  $z$  is not zero, said loop structure transforming unit is operable to generate a conditional statement for judging whether a loop count of an inner loop is  $y$  or  $z$  and to perform double looping transformation.

17. (Original) The program conversion device according to Claim 10,

wherein when an execution count of a loop is non-fixed, said loop structure transforming unit is operable to judge the execution count of the loop when the loop is executed and to perform double looping transformation so as to dynamically vary an iteration count in accordance with a judgment result.

18. (Original) The program conversion device according to Claim 10, further comprising a receiving unit operable to receive information showing that arrays are aligned to a cache line size,

wherein said instruction placing unit is operable to place a prefetch instruction in the loop, whose iteration count is x, for prefetching data stored one cache line ahead of data to be referenced within the iteration processing of the loop that is executed x number of times.

19. (Original) The program conversion device according to Claim 10, wherein said optimization directive information receiving unit is operable to receive information showing a relative position in a cache line, from which the array starts to access, said loop structure transforming unit is operable to perform the double looping transformation in accordance with the information.

20. (Original) The program conversion device according to Claim 10, wherein when the arrays are not aligned to the cache line size, said instruction placing unit is operable to place a prefetch instruction in the loop, whose iteration count is x, for prefetching data stored two cache lines ahead of data to be referenced within the iteration processing of the loop that is executed x number of times.

21. (Original) The program conversion device according to Claim 10,

wherein when the arrays are not aligned to the cache line size, said loop structure transforming unit is operable to judge a relative position in a cache line, from which the array starts to access, and operable to perform double looping transformation in accordance with a judgment result.

22. (Original) The program conversion device according to Claim 10, further comprising

a receiving unit operable to receive information that relates to a focused array,

wherein said loop structure transforming unit is operable to perform double looping transformation only on the focused array.

23. (Original) The program conversion device according to Claim 1,

wherein said loop structure transforming unit is operable to further perform double looping transformation on an outer loop, considering an innermost loop as one block.

24. (Currently Amended) A program conversion method for a processor which has an instruction set including an instruction that waits for a predetermined response from an outside source when the instruction is executed, comprising:

a step of performing double looping transformation so as to transform a structure of a loop, which is included in an input program and whose iteration count is  $x$ , into a nested structure where a loop whose iteration count is  $y$  is an inner loop and a loop whose iteration count is  $x/y$  is an outer loop; and

a step of converting the input program into an output program including the instruction by placing the instruction in a position outside the inner loop,

wherein the iteration count  $y$  of the input program in the inner loop is determined such that processing time of the input program in the inner loop constitutes all or part of latency time of the instruction placed outside the inner loop.

25. (Currently Amended) A record medium storing a program realizing a program conversion method for a processor which has an instruction set including an instruction that waits for a predetermined response from an outside source when the instruction is executed, the program causing a computer to execute:

a step of performing double looping transformation so as to transform a structure of a loop, which is included in an input program and whose iteration count is  $x$ , into a nested structure where a loop whose iteration count is  $y$  is an inner loop and a loop whose iteration count is  $x/y$  is an outer loop; and

a step of converting the input program into an output program including the instruction by placing the instruction in a position outside the inner loop,

wherein the iteration count  $y$  of the input program in the inner loop is determined such that processing time of the input program in the inner loop constitutes all or part of latency time of the instruction placed outside the inner loop.